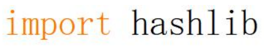
Merkel tree 实现实验报告

网络空间安全创新创业实践

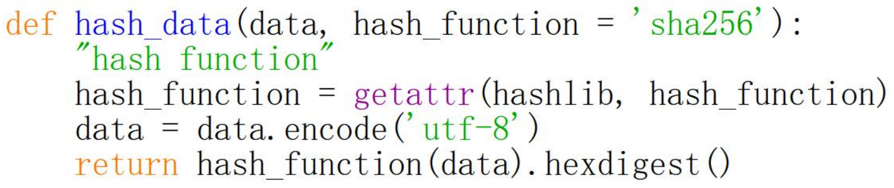
赵翔正 202000460090

导入 hashlib



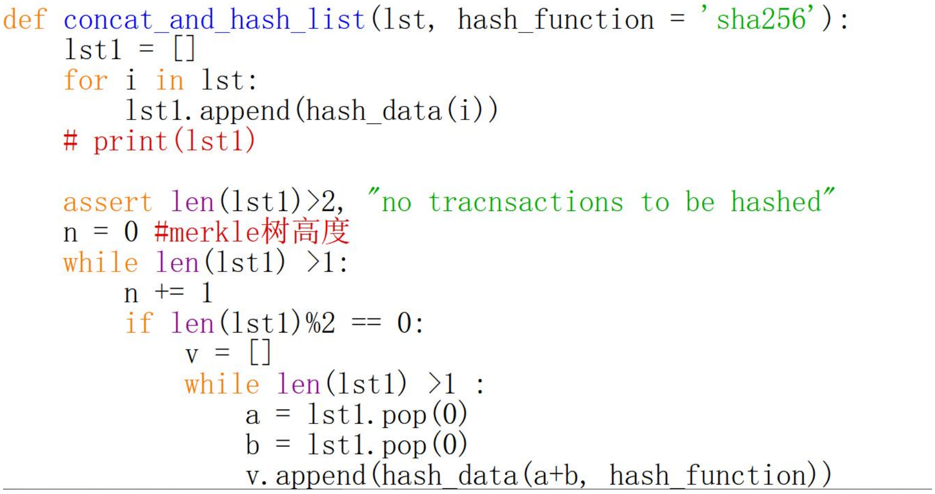
```
import hashlib
```

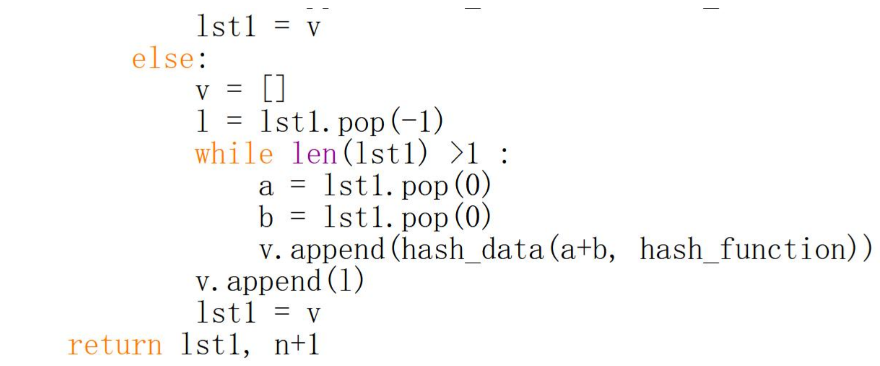
定义 hash 类型和编码方式



```
def hash_data(data, hash_function = ’sha256’):
    ″hash function″
    hash_function = getattr(hashlib, hash_function)
    data = data.encode(’utf-8’)
    return hash_function(data).hexdigest()
```

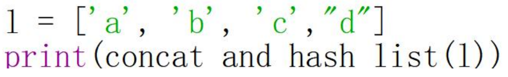
实现 merkel tree 的聚合过程





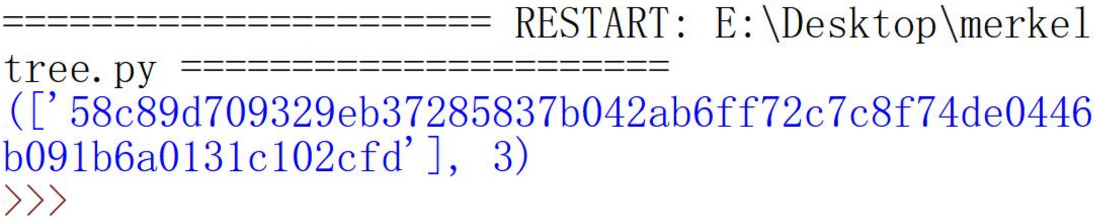
```
def concat_and_hash_list(lst, hash_function = ’sha256’):
    lst1 = []
    for i in lst:
        lst1.append(hash_data(i))
    # print(lst1)

    assert len(lst1)>2, ″no tracnsactions to be hashed″
    n = 0 #merkle树高度
    while len(lst1) >1:
        n += 1
        if len(lst1)%2 == 0:
            v = []
            while len(lst1) >1 :
                a = lst1.pop(0)
                b = lst1.pop(0)
                v.append(hash_data(a+b, hash_function))
            lst1 = v
        else:
            v = []
            l = lst1.pop(-1)
            while len(lst1) >1 :
                a = lst1.pop(0)
                b = lst1.pop(0)
                v.append(hash_data(a+b, hash_function))
            v.append(l)
            lst1 = v
    return lst1, n+1
```

检验是否正确



```
l = [’a’, ’b’, ’c’,″d″]
print(concat_and_hash_list(l))
```

结果如下



```
====================== RESTART: E:\Desktop\merkel
tree.py ======================
(['58c89d709329eb37285837b042ab6ff72c7c8f74de0446
b091b6a0131c102cfd'], 3)
>>>
```

其中“3”是 merkel tree 的高度